On storage codes allowing partially collaborative repairs

Liu, Shiqiu; Oggier, Frédérique

2014

Liu, S., & Oggier, F. (2014). On storage codes allowing partially collaborative repairs. 2014 IEEE International Symposium on Information Theory Proceedings, 2440-2444.

<https://hdl.handle.net/10356/104960>

<https://doi.org/10.1109/ISIT.2014.6875272>

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://dx.doi.org/10.1109/ISIT.2014.6875272>].

Downloaded on 26 Apr 2023 22:45:49 SGT

On Storage Codes Allowing Partially Collaborative Repairs

Shiqiu Liu and Frédérique Oggier

Division of Mathematical Sciences

Nanyang Technological University

Singapore

Email: SLIU012@e.ntu.edu.sg, frederique@ntu.edu.sg

Abstract—We consider the design of codes for distributed storage systems that are amenable to repair. We introduce the notion of partial collaboration to capture the property that nodes participating in a repair process may collaborate to different extents, bringing more nuances to the known cases: no collaboration or full collaboration during repair. We compute for this scenario the storage (per node) when the repair bandwidth (per node) is minimal, and conversely the repair bandwidth when the storage is minimal. We provide a generic code construction that enables repair of several failures through partial collaboration.

I. INTRODUCTION

When data is stored across a network of nodes, it is standard to also store a copy of it (or some other form of redundancy), to ensure that the data is not lost, should the node that stores it fail. If three copies of that data is kept on distinct nodes, this means (in terms of coding theory) that a repetition code of length 3 is used. There has been a lot of research done during these past years to replace repetition codes in distributed storage systems by more efficient erasure codes. More efficient here means first of all two things: (1) the storage overhead (or rate) of the code should be better than that of the repetition code of length 3, (2) the code should be amenable to *repair*. The notion of repair is not present in the design of classical erasure codes, it is a specificity of storage systems. Consider a codeword from some erasure code, and store one (or several) coefficients of this codeword across distinct network nodes. If some nodes fail, the original data will be protected by the erasure code, up to the code ability. However, if over time more nodes start failing, the data will be lost, and thus, before more failures occur, the system needs to trigger a repair process, that will make sure the data redundancy will be replenished, to keep the data protected over time. This is done by having nodes performing the repair contacting live nodes, downloading data from them, and computing the missing codeword coefficients. An efficient repair process may mean several things: on may desire it to be fast, to be computationally cheap, to require little repair bandwidth, or to contact a small number of live nodes only. One may refer to [1] for a survey of different code designs and constructions.

This paper focuses on the case where the distributed storage system has a threshold t of failures that it is willing to tolerate before triggering a repair, and this repair is subsequently done

by having t nodes collaborating, and exchanging data among each others. This idea was introduced independently in [2], [3], where a trade-off between the amount of data each node is storing, and the bandwidth needed to repair the t failures was studied. Two particular points were of interest: when the storage at each node is minimum, and when the repair bandwidth is minimum.

Definition 1: When an object of size M is stored across n nodes, with the property that any choice of k nodes should allow the object retrieval, the amount α of data stored per node has to be at least M/k . The regime where it is exactly M/k is called the *Minimum Storage Repair (MSR)* point. For a node to be repaired successfully, it is necessary that the repair bandwidth γ per node is at least α . The regime where $\alpha = \gamma$ is called the *Minimum Bandwidth Repair (MBR)* point.

When $t \geq 1$ failures are simultaneously repaired through collaboration, and every node participating in the repair process downloads data from d nodes, the total repair bandwidth γ_C per node (the subscript C refers to “collaboration”) is known to be [3], [2]:

$$\gamma_C = \begin{cases} \frac{M}{k} \frac{d+t-1}{d+t-k}, & \text{at MSR point} \\ \frac{M}{k} \frac{2d+t-1}{2d-k+t}, & \text{at MBR point.} \end{cases} \quad (1a)$$

$$\gamma_C = \begin{cases} \frac{M}{k} \frac{d+t-1}{d+t-k}, & \text{at MSR point} \\ \frac{M}{k} \frac{2d+t-1}{2d-k+t}, & \text{at MBR point.} \end{cases} \quad (1b)$$

In particular, for $t = 1$ failure (the subscript 1 refers to *one* failure), we have

$$\gamma_1 = \begin{cases} \frac{M}{k} \frac{d}{d+1-k}, & \text{at MSR point} \\ \frac{M}{k} \frac{2d}{2d-k+1}, & \text{at MBR point.} \end{cases} \quad (2a)$$

$$\gamma_1 = \begin{cases} \frac{M}{k} \frac{d}{d+1-k}, & \text{at MSR point} \\ \frac{M}{k} \frac{2d}{2d-k+1}, & \text{at MBR point.} \end{cases} \quad (2b)$$

Thus when $t \geq 1$ failures occur and the repair process is done without collaboration, the t failures are repaired successively, and we have a cost in bandwidth of γ_1 per node. This illustrates the gain brought by the collaboration process: both at MSR and MBR points, we have that

$$\gamma_C < \gamma_1 \iff (t-1)(1-k) < 0$$

which is true whenever $t \geq 2$ and $k \geq 2$.

The complete characterization of the region describing the trade-off between storage capacity and repair bandwidth has been computed in [4]. The design of codes for collaborative

Fig. 1. A data object $\mathbf{o} = (\mathbf{o}_1, \dots, \mathbf{o}_4, \dots)$ is encoded using a generator matrix G with columns $\mathbf{g}_i$. Node i stores $\mathbf{o}_i \mathbf{g}_i, \mathbf{o}_{i+1} \mathbf{g}_i, \mathbf{o}_{i+2} \mathbf{g}_i, \mathbf{o}_{i+3} \mathbf{g}_i$, $i = 1, 2, 3, 4$, and the first subscript is understood modulo 4. This figure shows the repair of $t = 4$ failures (nodes 1, 2, 3, 4): on the left, with no collaboration, on the right, with full collaboration (only the data uploaded by node 1 is shown, to not overload the figure). The two middle figures show two intermediate modes of collaboration. The more the nodes collaborate, the less they download from live nodes, and vice-versa.

repair is not easy, and different code constructions have been proposed for different regimes (e.g. [5],[6]). From a security point of view, the behavior of these codes have been studied both in the case of a passive adversary which eavesdrops [7], in which case the secrecy capacity of the codes has been considered, and of an active adversary, that is rogue nodes voluntarily corrupting the data that they transmit during repair [8]. It was shown in [8] that honest nodes are critical to the repair process, since the repair bandwidth obtained to secure (from an information theoretical point of view) cooperative codes from Byzantine attacks is worse than having no collaboration at all.

The contribution of this paper is to propose storage codes whose repair processes are done in a partially collaborative way: this means that the nodes participating in the repair process do collaborate by exchanging some data, but nodes do not communicate with all other repair nodes. This idea is illustrated in Figure 1, where $t = 4$ failures are being repaired in 4 different ways: on the left, every node is repaired only by downloading data from live nodes, on the right, the 4 repair nodes fully collaborate among each others (and download less from live nodes), and in the two middle figures, nodes share respectively one and two coefficient(s) that they own.

There are two main motivations to look at this scenario: (1) It provides a range of different regimes between the two known cases, no collaboration and full collaboration, allowing more flexibility for code designs. For example, one may want to improve a bit the repair bandwidth, but keep degrees of freedom for further optimizations. (2) From a security point of view, collaboration has been shown to be harmful [8]. One way to mitigate this threat could be to reduce the amount of collaboration, and trade the loss in repair bandwidth for honest nodes by a potential gain if some nodes get corrupted.

The approach we use is by now classical. We start by a min-cut analysis (similar to that of [4], [2], [9]) in Section II, which gives a bound on the amount of data that must circulate through the network, and in turn a constraint that relates the different parameters of interest (the storage capacity α and the repair bandwidth γ). We then study in Section III the two extreme regimes when the storage α and the repair bandwidth γ are respectively minimal. Finally in Section IV, we present a generic coding strategy which implements partially collaborative repairs. Design of optimal codes for the different trade-offs that are identified remains open.

II. A MIN-CUT BOUND

As done in [2], [4], [9], we consider an information flow graph, where the data flows from a source S to a data collector (DC). The repair process of t failures involves t nodes, which will all download β amount of data from d live nodes, and exchange β' amount of data with a subset of $t - s$ nodes, $1 \leq s \leq t$. The collaboration phase is assumed to be done such that every node has both incoming and outgoing degrees to be $t - s$ ($s = 1$ corresponds to full collaboration, and $s = t$ to none).

Theorem 1: Consider an information flow graph, where every node has a storage capacity of α , and repairs are performed by a group of t nodes, as described above. Suppose that a data collector DC connects to a subset of k nodes which were all involved in different phases of repairs, where each phase involves a group of u_i nodes, $1 \leq u_i \leq t$, and $k = \sum_{i=0}^{g-1} u_i$. Then a min-cut bound between the source S and the data collector DC is given by

$$\min cut(S, DC) \geq \min_{\mathbf{u} \in P} \left(\sum_{i \in \bar{I}} u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j) \beta\} + \sum_{i \in I} u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j) \beta + (t - s + 1 - u_i) \beta'\} \right) \quad (3)$$

where

$$I = \{i, t - s + 1 - u_i \geq 0\}, \quad \bar{I} = \{i, t - s + 1 - u_i < 0\}$$

and

$$P = \{\mathbf{u} = (u_0, \dots, u_{g-1}), 1 \leq u_i \leq t \text{ and } \sum_{i=0}^{g-1} u_i = k\}.$$

We will adopt the following notation. Repair nodes during the i th phase of repair are denoted by $x^{i,j}$, where j counts the nodes during the i th phase. Every node $x^{i,j}$ is seen a logical triple $(x_{in}^{i,j}, x_{coor}^{i,j}, x_{out}^{i,j})$ formed by an incoming node, a collaborating node, and an output node, to model the storage capacity and the collaborative process, as done in [2]. The main lines of the proof are similar to that of [2], but some details are nevertheless given for the sake of completeness.

Proof: Consider a data collector DC that connects to k output nodes corresponding to a set K of nodes, say $\{x_{out}^{i,j} : (i, j) \in K\}$. We show that any cut between S and DC in the graph has a capacity that satisfies (3). Since we may assume

that all the edges of the data collector have infinite capacity, we only consider the cut $(U, \bar{U})$, with $S \in U$ and $\{x_{out}^{i,j} : (i,j) \in K\} \in \bar{U}$. Let $\mathcal{C}$ denote the edges in the cut.

First Repair Phase. Let J be the set of indices such that $\{x_{out}^{0,j} : j \in J\}$ are the first output nodes in $\bar{U}$ corresponding to the first repair. The set contains $|\{x_{out}^{0,j} : j \in J\}| = u_0$ nodes. Consider a subset $M \subset J$ of size m such that $\{x_{in}^{0,j} : j \in M\} \subset U$ and $\{x_{in}^{0,j} : j \in J - M\} \subset \bar{U}$. Then m can take value between 0 and u_0 .

Consider firstly the m nodes $\{x_{in}^{0,j} : j \in M\}$. For each node $x_{in}^{0,j} \in U$, (1) either $x_{coor}^{0,j} \in U$, then $x_{coor}^{0,j} \rightarrow x_{out}^{0,j} \in \mathcal{C}$, and the contribution to the cut is α , (2) or $x_{coor}^{0,j} \in \bar{U}$, then $x_{in}^{0,j} \rightarrow x_{coor}^{0,j} \in \mathcal{C}$, and contribution to the cut is at least α .

Consider next the $u_0 - m$ other nodes $\{x_{in}^{0,j} : j \in J - M\}$. For each node the contribution comes from multiple sources. (1) The cut contains d edges carrying β coefficients: since $x_{out}^{0,j}$ are the first output nodes in $\bar{U}$, edges come from the output nodes in U . (2) When $t - s + 1 \geq u_0 - m$, the cut contains at least $t - s + 1 - u_0 + m$ edges carrying β' coefficients thanks to the coordination step: the node $x_{coor}^{0,j}$ has $t - s$ incoming edges $x_{in}^{0,k} \rightarrow x_{coor}^{0,j}$. However, since $|\{x_{in}^{0,k}\}| \cap \bar{U} = u_0 - m - 1$, the cut contains at least $(t - s) - (u_0 + m - 1)$ such edges. (3) When $t - s + 1 < u_0 - m$, the least number of edges carrying β' will be 0, that is the cut contains no edges carrying β' .

Therefore, the total contribution of those nodes when $t - s + 1 \geq u_0 - m$ is $c_0(m) \geq m\alpha + (u_0 - m)(d\beta + (t - s + 1 - u_0 + m)\beta')$. When $t - s + 1 < u_0 - m$, $c_0(m) \geq m\alpha + (u_0 - m)d\beta$. Since the function c_0 is concave on the interval $[0, u_0]$, Jensen's inequality yields

$$c_0(m) \geq u_0 \min\{\alpha, d\beta + (t - s + 1 - u_0)\beta'\},$$

or $c_0(m) \geq u_0 \min\{\alpha, d\beta\}$.

Second Repair Phase. Let $\{x_{out}^{1,j} : j \in J\}$ be the second output nodes in $\bar{U}$ corresponding to a second repair. We repeat the same reasoning. We firstly consider the m nodes $\{x_{out}^{1,j} : j \in M\} \subset U$, where the contribution of each node is α .

Then, we consider the $u_1 - m$ nodes $\{x_{out}^{1,j} : j \in U - M\}$. For each node, we have (1) the cut contains at least $d - u_0$ edges carrying β : since these $\{x_{out}^{1,j}\}$ are the second output nodes in $\bar{U}$, at most u_0 edges come from output nodes in U , so at least $d - u_0$ edges come from output nodes in $\bar{U}$, (2) and similarly to the first phase, the cut contains at least $t - s + 1 - u_1 + m$ edges carrying β' when $t - s + 1 \geq u_1 - m$; the cut contains no edges carrying β' , when $t - s + 1 < u_1 - m$.

Thus the total contribution of these nodes when $t - s + 1 \geq u_1 - m$ is $c_1(m) \geq u_1 \min\{\alpha, (d - u_0)\beta + (t - s + 1 - u_1)\beta'\}$, and $c_1(m) \geq u_1 \min\{\alpha, (d - u_0)\beta\}$ if $t - s + 1 < u_1 - m$.

In general, we have for the i th repair phase that

$$c_i(m) \geq u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j)\beta + (t - s + 1 - u_i)\beta'\},$$

when $t - s + 1 \geq u_i - m$, and when $t - s + 1 < u_i - m$,

$$c_i(m) \geq u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j)\beta\}.$$

Summing these contributions leads to (3). $\blacksquare$

Using the Minimum cut-Maximum flow Theorem, we get that the initial file size M must satisfy

$$M \leq \min_{\mathbf{u} \in P} \left(\sum_{i \in I} u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j)\beta + (t - s + 1 - u_i)\beta'\} + \sum_{i \in \bar{I}} u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j)\beta\} \right) \quad (4)$$

where

$$I = \{i, t - s + 1 - u_i \geq 0\}, \quad \bar{I} = \{i, t - s + 1 - u_i < 0\}$$

and

$$P = \{\mathbf{u} = (u_0, \dots, u_{g-1}), 1 \leq u_i \leq t \text{ and } \sum_{i=0}^{g-1} u_i = k\}.$$

Corollary 1: When $s = 1$, the collaboration phase involves all the other $t - s = t - 1$ nodes, and

$$M \leq \sum_{i=0}^{g-1} u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j)\beta + (t - u_i)\beta'\},$$

for $1 \leq u_i \leq t$ such that $\sum_{i=0}^{g-1} u_i = k$ which is the known bound from [3], [2].

Proof: Set $s = 1$ in (4) to get

$$M \leq \min_{\mathbf{u} \in P} \sum_{i \in I} u_i \min\{\alpha, (d - \sum_{j=0}^{i-1} u_j)\beta + (t - u_i)\beta'\}$$

since $\bar{I} = \{i : t - u_i < 0\}$ is now empty ($u_i \leq t$), while $I = \{i : t - u_i \geq 0\} = \{0, \dots, g-1\}$ contains all the possible indices for i . $\blacksquare$

III. MINIMUM STORAGE AND BANDWIDTH POINTS

a) The Minimum Storage Repair (MSR) Point: At MSR point, we have $\alpha = \frac{M}{k}$. The highest contribution from β comes when there is no contribution in β' , that is when $u_i = t - s + 1$ for all i . Then $g = k/(t - s + 1)$, $\bar{I}$ is empty, and (4) with equality becomes

$$\sum_{i=0}^{k/(t-s+1)-1} (t - s + 1) \min(\alpha, (d - i(t - s + 1))\beta) = M. \quad (5)$$

Since $\alpha = M/k$, each term in this sum is at most $(t - s + 1)\alpha$, and since there are $k/(t - s + 1)$ terms in this sum, it must be that

$$(t - s + 1) \min(\alpha, (d - i(t - s + 1))\beta) = \frac{M(t - s + 1)}{k}$$

that is

$$(d - i(t - s + 1))\beta \geq \frac{M}{k},$$

for $i = 0, \dots, \frac{k}{t-s+1} - 1$. When $i = \frac{k}{t-s+1} - 1$, we get

$$\beta = \frac{M}{k} \frac{1}{d - k + t - s + 1}.$$

Fig. 2. The total repair bandwidth γ as a function of $1 \leq s \leq 4$ for $t = 4$, for $k = 6$ (line below), and $k = 8$ (line above). The points at $s = 1$ correspond to the full collaboration case, while those at $s = 4$ correspond to no collaboration.

Conversely, the highest contribution from β' comes when $u_i = 1$ for all i . Then $g = k$, and i ranges from 0 to $k - 1$. Then (4) with equality gives

$$\sum_{i=0}^{k-1} \min(\alpha, (d-i)\beta + (t-s)\beta') = M \quad (6)$$

and using the same argument as above, we now need

$$(d-i)\beta + (t-s)\beta' \geq \frac{M}{k}.$$

Then we get

$$\begin{aligned} \beta' &\geq \frac{1}{t-s} \left(\frac{M}{k} - \beta(d-k+1) \right) \\ &= \frac{1}{t-s} \frac{M}{k} \left(1 - \frac{d-k+1}{d-k+t-s+1} \right) \\ &= \frac{M}{k} \frac{1}{d-k+t-s+1} \end{aligned}$$

and the repair bandwidth is

$$\begin{aligned} \gamma &= d\beta + (t-s)\beta' \\ &= \frac{M}{k} \frac{d+t-s}{d-k+t-s+1}. \end{aligned}$$

To summarize, we have at MSR point

$$\boxed{\alpha = \frac{M}{k}, \gamma = \frac{M}{k} \frac{d+t-s}{d-k+t-s+1}, \beta = \beta' = \frac{M}{k} \frac{1}{d-k+t-s+1}.}$$

As a sanity check, we notice that $s = 1$ indeed gives the known formulas ([2], [4])

$$\beta_C = \beta'_C = \frac{M}{k} \frac{1}{d-k+t} \quad (7)$$

and (1a). The optimal total repair bandwidth is illustrated in Figure 2, compared with the known cases where either full collaboration or no collaboration is done.

b) *The Minimum Bandwidth Repair (MBR) Point:* To optimize γ , we let α grow, and (5) gives

$$(t-s+1) \sum_{i=0}^{k/(t-s+1)-1} (d-i(t-s+1))\beta = M,$$

that is

$$\beta = \frac{M}{k} \cdot \frac{2}{2d-k+t-s+1}.$$

We next compute β' . From (6), we get

$$\sum_{i=0}^{k-1} ((d-i)\beta + (t-s)\beta') = M,$$

thus

$$\beta' = \frac{M}{k} \frac{1}{2d-k+t-s+1}.$$

Finally, at MBR point, the total repair bandwidth $\gamma = d\beta + (t-s)\beta'$ is equal to the storage capacity α , and

$$\gamma = \frac{M}{k} \frac{2d+t-s}{2d-k+t-s+1}.$$

To summarize, at MBR point, we have

$$\boxed{\alpha = \gamma = \frac{M}{k} \frac{2d+t-s}{2d-k+t-s+1}, \beta = \frac{M}{k} \frac{2}{2d-k+t-s+1}, \beta' = \frac{M}{k} \frac{1}{2d-k+t-s+1}.}$$

Another sanity check shows that we indeed recover the known formulas ([2], [4])

$$\beta = \frac{M}{k} \cdot \frac{2}{2d-k+t}, \beta' = \frac{M}{k} \frac{1}{2d-k+t}$$

and (1b) when $s = 1$.

IV. A GENERIC CODE CONSTRUCTION

We propose a code construction which is a slight generalization of that of [3] for full collaborative repair.

Consider an object $\mathbf{o}$ of length kt with coefficients in the finite field $\mathbb{F}_q$, and represent it as a matrix $\mathbf{O}$ in $\mathbb{F}_q^{t \times k}$, that is

$$\mathbf{O} = \begin{bmatrix} o_{1,1} & \dots & o_{1,k} \\ \vdots & & \vdots \\ o_{t,1} & \dots & o_{t,k} \end{bmatrix} = \begin{bmatrix} \mathbf{o}_1 \\ \vdots \\ \mathbf{o}_t \end{bmatrix}$$

where $\mathbf{o}_i$ is the i th row of $\mathbf{O}$, $i = 1, \dots, t$. Let $G \in \mathbb{F}_q^{k \times n}$ be the generator matrix of an MDS code, whose columns are denoted by $\mathbf{g}_i$, $i = 1, \dots, n$

$$G = [\mathbf{g}_1, \dots, \mathbf{g}_n].$$

The i th node stores $\mathbf{Og}_i$.

Object Retrieval. If any k nodes are contacted, then the MDS property of G ensures that the object is recovered.

Repair. The system waits for t failures to have happened, before starting the repair process. Without loss of generality, we label these t nodes from 1 to t . Then t live nodes each contact $d = k$ nodes, say the i th node among those t live nodes connect to nodes $i_1, \dots, i_k$ and downloads

$$\mathbf{o}_i \mathbf{g}_{i_j}, \mathbf{o}_{i+1} \mathbf{g}_{i_j}, \dots, \mathbf{o}_{i+s-1} \mathbf{g}_{i_j} \in \mathbb{F}_q, \quad j = 1, \dots, k,$$

where the indices are understood modulo t (ranging from 1 to t). Again using the MDS property of G , the i th node computes $\mathbf{o}_j \mathbf{g}_i$, $j = i, \dots, i + s - 1$, and after the download phase, it can compute $\mathbf{o}_j \mathbf{g}_i$, $j = i, \dots, i + s - 1$ and thus has obtained s of the t coefficients it needs to store. The $t - s$ missing coefficients are obtained through (partial) collaboration as follows: the i th node contacts nodes $i - 1, \dots, i - (t - s)$. It already owns $\mathbf{o}_i \mathbf{g}_i, \dots, \mathbf{o}_{i+s-1} \mathbf{g}_i$, and gets $\mathbf{o}_{i-1} \mathbf{g}_i, \dots, \mathbf{o}_{i-(t-s)} \mathbf{g}_i$ from the other nodes.

Note that the code is such that it allows repairs in parallel, no matter the level of collaboration. This coding strategy is illustrated for $t = 4$ and $1 \leq s \leq 4$ in Figure 1.

Lemma 1: The above construction provides a storage code with partial collaboration, with a storage capacity of $\alpha = t$, download bandwidth $\beta = s$, exchange bandwidth $\beta' = 1$ and total repair bandwidth

$$\gamma = sk + (t - s), \quad 1 \leq s \leq t,$$

for a repair degree of $d = k$, and collaborative degree of $t - s$.

Proof: The size of the object is tk , thus $\alpha = tk/k = t$ corresponds to the minimum storage value that α can have. During repair, every node contacts $d = k$ live nodes, and downloads $\beta = s$ coefficients from them, accounting for ks , after which $t - s$ coefficients are exchanged, for a total of $sk + (t - s)$. ■

Corollary 2: If $s = 1$, the exchange process involves $t - s = t - 1$ repair nodes, this is the collaborative case with

$$\gamma = \gamma_C = k + (t - 1), \quad \beta_C = \beta'_C = 1.$$

If $s = t$, then there is no collaboration, and the repair process performs t repairs of one failure:

$$\gamma = \gamma_1 = tk, \quad \beta_1 = t.$$

Proof: Indeed, set $M = kt$, $d = k$ in (1a) for γ_C and in (7) for $\beta_C = \beta'_C$. Again, set $M = kt$, $d = k$ in (2a) for $t\gamma_1$. From (7) with $t = 1$ gives

$$\beta_1 = \frac{M}{k} \frac{1}{d - k + 1}.$$

Choose $M = kt$ and $d = k$. ■

Recall that $d = k$, $\alpha = t$ and $M = tk$. Let us compare the code parameters with those found above at MSR point, since $\alpha = t = M/k$.

$$\begin{aligned} \gamma &= \frac{M}{k} \frac{d + t - s}{d - k + t - s + 1} = \frac{t(k + t - s)}{t - s + 1}, \\ \beta &= \beta' = \frac{M}{k} \frac{1}{d - k + t - s + 1} = \frac{t}{t - s + 1}. \end{aligned}$$

Assume for now that $s \neq 1$, and $k \geq 2$. We notice that:

- $\beta = \frac{t}{t-s+1} \leq s \iff t \geq s$, with equality when $t = s$,
- $\beta' = \frac{t}{t-s+1} \geq 1 \iff s \geq 1$, with equality when $s = 1$,
- $\gamma = \frac{t(k+t-s)}{t-s+1} \leq ks + t - s \iff t \geq s$, with equality when $t = s$.

The case $s = 1$ is discussed in Corollary 2. This shows all together that the generic construction that we proposed

Fig. 3. The total repair bandwidth γ as a function of $1 \leq s \leq 4$ for $k = 6$ and $t = 4$. The straight line corresponds to the generic coding strategy, while the other line is the optimal strategy.

achieves the optimal total repair bandwidth γ if and only if $s = 1$ or $s = t$, that is the full collaboration case, or the case with no collaboration. For all other values of s , the download bandwidth β and thus the overall γ is higher than the optimal one. The gap between the proposed generic strategy and the optimal one given by the min-cut bound analysis is shown in Figure 3. Constructions of optimal codes thus remain open.

ACKNOWLEDGEMENT

The research of S. Liu is supported by the Singapore National Research Foundation under Research Grant NRF-RF2009-07. The research of F. Oggier for this work is supported by the MoE Tier-2 grant “eCODE: Erasure Codes for Datacenter Environments”.

REFERENCES

- [1] F. Oggier, A. Datta, “Coding Techniques for Repairability in Networked Distributed Storage Systems”, in the series *Foundations and Trends in Communications and Information Theory*, vol. 9, no. 4, June 2013.
- [2] A.-M. Kermarrec, N. Le Scouarnec, G. Straub, “Repairing Multiple Failures with Coordinated and Adaptive Regenerating Codes”, *The 2011 International Symposium on Network Coding (NetCod 2011)*.
- [3] K. Shum, “Cooperative Regenerating Codes for Distributed Storage Systems”, *The International Conference on Communications (ICC 2011)*, Kyoto. Available at <http://arxiv.org/abs/1101.5257>
- [4] K. W. Shum, Y. Hu, “Cooperative Regenerating Codes”, preprint, available at <http://arxiv.org/abs/1101.5257>
- [5] K. W. Shum and Y. Hu, “Exact Minimum-Repair-Bandwidth Cooperative Regenerating Codes for Distributed Storage Systems”, *The International Symposium on Information Theory (ISIT 2011)*, Saint-Petersburg.
- [6] N. Le Scouarnec, “Exact scalar minimum storage coordinated regenerating codes”, *The International Symposium on Information Theory (ISIT 2012)*, Boston. Available at <http://arxiv.org/abs/1202.0457>
- [7] O. O. Koyluoglu, A. S. Rawat, S. Vishwanath, “The secrecy capacity of minimum bandwidth cooperative regenerating codes”, *The International Symposium on Information Theory (ISIT 2013)*, Istanbul. Available at <http://arxiv.org/abs/1210.3664>.
- [8] F. Oggier, A. Datta, “Byzantine Fault Tolerance of Regenerating Codes”, *The 11th IEEE International Conference on Peer-to-Peer Computing (P2P 2011)*, Kyoto. Available at <http://arxiv.org/abs/1106.2275>.
- [9] A.G. Dimakis, P.B. Godfrey, Y. Wu, M.J. Wainwright, K. Ramchandran, “Network Coding for Distributed Storage Systems”, *IEEE Trans. on Information Theory*, vol. 56, no. 9, Sept. 2010.